



Sobreescritura de métodos

Transcripción

[00:00] Y como no podía ser de otra forma, si queremos validar que lo que hemos hecho está bien hecho, vamos a crear nuestro test. Para eso lo que tenemos que hacer es aquí new class y creamos aquí nuestro TestCuenta. Perfecto. Le damos a finish, tenemos nuestra clase, creamos nuestro método main, que ya sabemos, escribimos main, "Ctrl + espacio", enter, y genera automáticamente nuestro método main.

[00:33] Y vamos a crear una CuentaCorriente cc igual new CuentaCorriente, con agencia número 1 y número de cuenta número 1, vamos a ponerle solamente para fines del ejemplo. Punto y coma, vamos a crear otra cuenta, pero esta va a ser una cuenta de ahorros. Y vemos que está aquí abajo, "Ctrl + espacio", cuenta de ahorros, CuentaAhorros igual new "Ctrl + espacio", cuenta de ahorros, y esta va a ser 2, 2. 2,3. Número de agencia y número de cuenta.

[01:18] Solamente por fines del ejemplo. Vamos ahora a hacer un depósito a cada cuenta. Vamos a ponerle que la cuenta corriente vamos a depositar, vemos que tenemos acceso a todos los métodos de la clase padre, y le vamos a depositar 2000. Y ahora, tenemos una nueva regla de negocios que acaba de salir en este momento y es que para cada transferencia que yo haga entre las cuentas de ahorro para cuenta corriente, yo voy a cobrar una comisión de 20 centavos. ¿Qué les parece?

[01:58] Entonces, si yo ahorita deposito a mi cuenta corriente 2000 y le digo que la cuenta corriente que transfiera el valor digamos transfiero 1000 a la cuenta de destino, que es la cuenta corriente, perdón, la cuenta de ahorros, si yo voy

ver el saldo, vamos a darle un Sysout, para imprimir el saldo. GetSaldo. Vamos a ejecutar este código, me dice que lo guarde, le digo okay. Y vemos que da saldo me da 1000.

[02:47] Bajamos un poco más. Y en la cuenta de ahorros si imprimo el saldo, ejecutamos, guardamos, y me da 1000. Aquí vemos que tengo 1000 en cada cuenta. Pero entonces ahora ¿cuál es mi detalle? Si yo necesito implementar esa lógica de negocio, de cada transferencia de cuenta corriente a cuenta de ahorros, yo como banco voy a ganar 20 centavos de esa transferencia que van a ser pagados por la cuenta que está haciendo transferencia que en este caso es la cuenta corriente.

[03:26] ¿Cómo yo puedo implementar ese método? Aquí entra lo que es la sobreescritura de métodos. Entonces vamos a lo que es nuestra cuenta corriente, y aquí nosotros podemos sobrecribir el método saca. ¿Por qué el método saca? Porque si vamos a nuestra clase cuenta el método transfiere, vamos a recordar un poco lo que ya hemos hecho.

[04:00] Si vemos aquí el método transfiere, vemos que él usa el método saca del valor para después depositar en el destino. Y entonces, ya que es él quien va a pagar esa comisión, entramos a su método, control, y vemos aquí claramente que lo que él aquí hace es restar ese valor del saldo actual de la cuenta. Pero él solamente resta el valor y no resta nuestra comisión.

[04:39] Si nosotros simplemente vamos directamente al método aquí, en cuenta, vamos a hacer que para cada cuenta que herede de esta clase principal cuenta use exactamente la misma red de negocios, que no necesariamente va a ser verdad siempre. ¿Por qué? Porque en algunos casos cuando sea cuenta de ahorros de la que yo esté haciendo la transferencia, mi comisión puede ser mayor.

[05:02] Justamente para evitar esto, voy a dejar el método aquí tal como está, limpio, y lo voy a editar en cuenta corriente. ¿Y cómo lo hago? Si yo escribo

aquí saca y le doy "Ctrl + espacio", automáticamente el IDE ya me está diciendo: "¿De casualidad tú no quieres sobrecribir el método que ya existe en cuenta?" Entonces selecciono y me va a autogenerar este código que ya lo vimos en la clase anterior.

[05:30] ¿Qué significa este código? Override es la anotación que me indica que este método está siendo sobreescrito de la clase padre y que no es un método nuevo. Y bueno, esto es lo que indica que fue autogenerado. Y él por defecto aquí está sacando el valor de lo que le estamos pasando, que es el método por defecto que está implementado. Nuevamente. ¿Override qué nos indica?

[06:00] Que este método está sobreescribiendo el método de su clase padre, que está aquí. Volvemos a este método y ahora vamos a agregarle pues la comisión. Para eso vamos a ponerle un double comisión, hay un type aquí, listo, y a la comisión le vamos a agregar un 0.2, de 20 centavos que vamos a cobrar por cada transacción de retiro.

[06:38] Entonces ahora, saca valor más, tengo que sacar el valor. ¿Mas qué? Mas mi comisión. "Ctrl + espacio" comisión. Ahora, cada vez que yo desde la cuenta corriente realice un saque o una transferencia, voy a cobrar mi comisión mas el valor que estoy transfiriendo. Vamos a comprobarlo y ejecutamos nuevamente este código y vemos aquí que ahora sí yo cobré mi comisión. 999.80 y aquí 1000.

[07:12] Entonces, esto es lo que se llama como polimorfismo. Una cuenta se volvió cuenta corriente y la cuenta corriente puede implementar su propia lógica de negocio basada en su clase cuenta, en su clase padre, sobreescribiendo datos importantes sobre la escritura de métodos. La firma del método, la firma del método es esto, nombre, tipo de método, parámetros, tienen que ser exactamente iguales.

[07:47] Si yo hago esto, si yo acá en vez le pongo un entero, entonces me va a dar un error. ¿Por qué? Porque no va a ser un método sobreescrito. Yo tendría

que borrar esta línea y él va a compilar ahí. ¿Por qué? Porque este método no está sobreescribiendo a su padre, este método es un método nuevo.

[08:04] Yo tendría que tener exactamente el mismo tipo de dato, el mismo parámetro, mismo nombre de método porque si yo acá también le pongo sacar por ejemplo, también me va a dar un error porque no es el mismo método del padre. Eso es el primer dato qv tienen que tener en cuenta.

[08:22] El segundo que es esta anotación, es muy importante, no la borren. ¿Por qué? Porque si ustedes borran esta anotación, este método no va a sobrecribir el método del padre, este método va a ser considerado un método nuevo de esta clase y no es siempre lo que nosotros queremos. ¿Por qué?

[08:38] Porque si por ejemplo yo aquí leí en la clase padre método saca y yo entendí sacar y hago esto, sacar, él va a compilar, no me va a decir absolutamente nada. Voy a implementarlo, y cuando haga el test, cuando ejecute el código, voy a guardar, él va a seguir guardo, 1000, 1000. ¿Por qué? Porque él no está sobreescribiendo el método de su padre, él está ejecutando un método propio y ahí comienzan los problemas, cuando se trata pues de sobreescritura y métodos.

[09:11] Entonces una recomendación muy buena, anotación override. Tiene que estar aquí. Esto me asegura, me confirma siempre que este método lo estoy sobreescribiendo y que no estoy cometiendo ningún error. Como les digo siempre, practiquen esto, estos conceptos son muy importantes y los van a ver muy, muy, muy seguido.

[09:36] Practíquenlo mucho, traten de crear más variables de cuenta y cualquier pregunta o cualquier duda que tengan, ya saben que tienen ahí el foro. Nos vemos en la siguiente clase.

